Presented at The 13th International Workshop on Spoken Dialogue Systems Technology, IWSDS 2023, Los Angeles, USA, University of Southern California Institute for Creative Technologies, February 21-24, 2023

Batch-level GANs to promote dialogue response variety

Asier López Zorrilla, Alain Vázquez and M. Inés Torres

Abstract Exploiting large pretrained transformers has become one of the most popular approaches for dialogue modelling. Nonetheless, due to their lack of robustness and explainability, we believe it is necessary to keep exploring alternative methodologies. In this work, we focus on Generative Adversarial Networks (GANs) for open-domain dialogue generation. In particular, we extend the idea of conventional GAN discriminators, which operate at a single response level, to the batch level. Our proposed discriminator evaluates how human a set of responses are for the corresponding dialogue contexts. We show that batch-level GANs outperform response-level GANs and a MLE baseline in terms of variability, without hurting the semantic coherence, according to our metrics. We believe that our proposal could benefit future work in GAN-related research, as well as other AI systems that employ discriminators.

1 Introduction

Large pretrained language models such as GPT-2 [5], BERT [8], and more recently GPT-3 [4] or PaLM [7], have surely supposed an important milestone in Natural Language Processing (NLP). They allow to achieve state-of-the-art (SOTA) results (or close to them) in many NLP tasks, including dialogue generation [20], by only fine-tuning the models with domain-specific data [39, 1], or even without re-training them at all [29]. This approach is very attractive for researchers and developers, because it allows building fully operational systems without having to worry too much about hyper-parameter or architectural choices. Furthermore, similar pretrained models have also proven successful in other Artificial Intelligence areas, such as Audio or Image Processing [6, 46]. Nonetheless, there is an in-

Asier López Zorrilla, Alain Vázquez and M. Inés Torres
University of the Basque Country UPV/EHU, e-mail: {asier.lopezz; alain.vazquez; manes.torres}@ehu.eus

creasing concern in the research community regarding the overuse of such models, since the decisions they make are hardly explainable and not always robust [53, 12]. Thus, we believe it is necessary to keep exploring alternative NLP methodologies which could potentially avoid such issues and/or complement current SOTA models.

In this work, we focus on Generative Adversarial Networks (GANs) [11] for open-domain dialogue generation. The learning methodology for GANs involves training two neural networks, a generator and a discriminator, in an adversarial fashion. The generator tries to learn a data distribution while the discriminator learns whether a given sample corresponds to the training data or has been generated by the generator. In the context of dialogue systems, the generator produces a response given the dialogue context, whereas the discriminator acts as a kind of automatic Turing Test. It is noteworthy that more modern approaches to NLP, such as BERT, also include a discriminator as part of their neural network, which has proven to be very beneficial to both Question Answering and Natural Language Inference tasks. GPT-2-based dialogue models often make use of such a feature too [50, 14, 37]. Thus, we believe that systems based on these transformers should also benefit from novel GAN/discriminator ideas such as the ones we propose next.

GANs were initially employed in this area in order to alleviate the lack of generation variability of sequence-to-sequence neural network models [49, 43, 42, 21]. Such neural models have usually been learnt from corpora composed of dialogue context-response pairs, via Supervised Learning minimising the token-level cross-entropy loss, a method often called MLE [49]. In this framework, the neural network is trained to minimise a distance between the generated response and the desired one. One reason why this methodology frequently yields models that tend to generate dull and safe responses is that it does not take into account the *one-to-many* property of conversational input-output pairs [47]. Conversely, GANs allow many correct outputs for the same input, which is very convenient for dialogue generation [22, 44, 47]. In this case, the discriminator may judge many responses as valid given the same context.

However, there are still limitations in the way such discriminators operate. To the best of our knowledge, almost every discriminator in related works operates at the response-level [52, 26, 15, 38, 32, 31]. This is, they evaluate how appropriate a single response is given a dialogue context. We propose to provide the discriminators with a wider view of the generator's behaviour. We name our proposal batch-level discriminators. They evaluate a set of responses given a set of dialogue contexts. Thus, they are less likely to be fooled by complex yet repetitive or not very informative responses.

The rest of the paper is organised as follows. Section 2 presents related works. We provide details about the GAN framework in Section 3. We present our batch-level GAN in Section 4, and we show that they outperform both the MLE baseline and the response-level GAN in our experiments, in Sections 5 and 6. Last, we end with some concluding remarks in Section 7.

2 Related work

Increasing the variety of neural dialogue models. The lack of variety and the non-informativeness of neural dialogue models have been tackled in several ways, apart from GANs and the aforementioned systems based on pretrained transformers. One family of solutions propose modifying the training objective to avoid the limitations of MLE. Among others, MMI-based objective functions [21], *bag-of-words* losses [55], frequency-aware losses [16], negative training [23], and backward reasoning [25] have been explored.

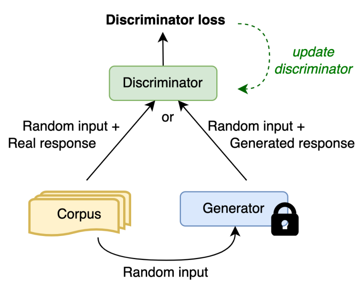
Another hypothesis of why sequence-to-sequence models end up producing generic and dull responses holds that the dialogue history alone might not be sufficient for producing informative responses. To alleviate this, additional information about the topic of the conversation can be included in the models [51, 45], Wikipedia or similar sources can be read to condition the response generation [10], and Internet search queries can be learnt and responses generated based on the search results [18].

Dialogue-generating GANs. The first works on dialogue-generating GANs were motivated by [52], which proposed a Reinforcement Learning-based approach to train text-based GANs. Specifically, [22] were one of the first to use GANs to develop dialogue models. Then, many others tried to improve on this baseline. [26] propose to use discriminators that rank input-output pairs, instead of solely performing a *real/fake* binary classification. [44] extend the generator to allow multi-turn dialogue generation, using a hierarchical Recurrent Neural Network (RNN). [15] propose an extension of the REINFORCE-based objective function to regularise the generative model. [38] show how to leverage the novel Cascade GANs to model the relations between sentences with Graph Convolutional Networks. Last, [47] present step-GANs, which allow different rewards for each generated token. Similarly, [32] presented a non-RL-based approach to train dialogue GANs, which also produces different gradients for each token.

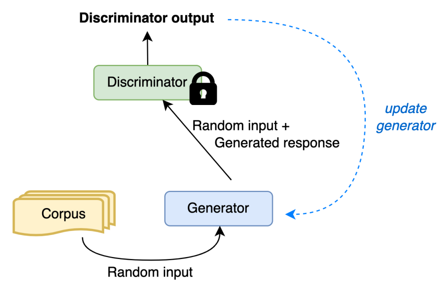
Batch-level GANs. Batch or minibatch discrimination was first proposed by [40], even if in a very different form than our proposal. They compute/learn a set of handcrafted batch-level statistics and include it in a layer at the end of the discriminator. [17] propose a similar strategy; to compute the standard deviation of input features and feed it to the last discriminator layer. However, these two discriminators still work over single samples, rather than batches. Closer to our GAN architecture, [35] present a permutation-invariant discriminator architecture that processes sets of instances. They propose to train their discriminator with mixed batches of real and fake samples, and to predict the ratio of real instances. These works show the potential of batch-level discrimination in the image generation task. Nonetheless, we are not aware of any similar work for dialogue-related tasks.

3 The GAN framework

The methodology to train a dialogue system in the GAN framework involves iteratively updating the generator and the discriminator. The generator is trained to fool the discriminator and make it predict that its responses are human-like, and in contrast the discriminator is trained to distinguish between human and bot responses. In most AI areas, this is done with two optimisation procedures: 1) the discriminator is trained to discriminate between samples generated by the generator and sampled from a corpus; and 2) the generator is trained to minimise the output of the discriminator. This process is illustrated in Figure 1.



(a) Optimisation of the discriminator.

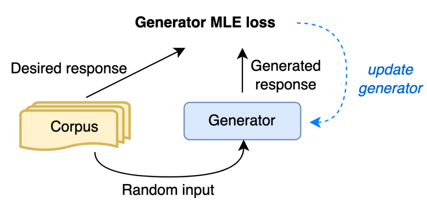


(b) Optimisation of the generator.

Fig. 1: Main two steps for the GAN optimisation procedure. The lock indicates when the parameters of the networks are frozen.

Additionally and specifically for the dialogue task, a third optimisation step is usually introduced in order to make the whole optimisation process more stable [22, 15]. It consists in performing a MLE of the parameters of the generator to predict the response in the corpus, as represented in Figure 2.

Fig. 2 MLE of the parameters of the generator via SL.



Baseline Discriminator. We compare the proposed batch-level discriminator (see Section 4) with a baseline discriminator. The typical task attributed to discriminators is to judge how natural or human a response is given the dialogue history. Our baseline discriminator is composed of two deep bidirectional LSTM. One is devoted to processing the dialogue history x, and the other one, is the response

r. Both x and r integer sequences are converted to word vector sequences via a word vector matrix $\mathbf{W}$. Then, encoder outputs are concatenated and processed by a standard MLP. This outputs a scalar between 0 and 1 (using a sigmoid activation function), which indicates the probability of r being produced by a bot. In other words, it should output values closer to 0 if r was present in the corpus, and closer to 1 otherwise. During training, the discriminator's parameters are updated to minimise a binary cross-entropy loss:

$$L_D = \frac{1}{|\mathscr{B}_D|} \sum_{x,r,l \in \mathscr{B}_D} -\left[l \cdot \log a + (1-l) \cdot \log(1-a)\right] , \tag{1}$$

where $\mathscr{B}_D$ is a batch composed of tuples of input utterances x, responses r and boolean labels l indicating if r was sampled from the corpus ($l = 0$) or generated by the generator ($l = 1$), and a the output of the network given x and r.

Generator: MLE training. In the GAN framework for dialogue, the generator is in charge of generating the system or bot's response given the dialogue history. In our experiments, we restrict the dialogue history to the last turn, because we found it challenging enough to make the GAN converge this way. Our generator is a RNN sequence-to-sequence network with attention [2]. A MLE of the parameters of the generator is carried out by minimising the token level cross-entropy loss L_{MLE} (Figure 2):

$$L_{MLE} = \frac{1}{|\mathscr{B}_{MLE}|} \sum_{x,s \in \mathscr{B}_{MLE}} \frac{1}{|s|} \sum_{t=1}^{|s|} -\log \mathbf{p}_t[s_t] , \tag{2}$$

where $\mathscr{B}_{MLE}$ is a batch composed of pairs of inputs x and desired outputs s sampled from the training data, s_t each of the words in s, and $\mathbf{p}_t[s_t]$ the output of the network in the t-th time step corresponding to the token s_t.

Adversarial training of the generator. The adversarial loss for the generator is the output of the discriminator (Figure 1b), after the latter is fed with a batch of input utterances and the responses of the generator to those same input utterances:

$$L_G = \frac{1}{|\mathscr{B}_G|} \sum_{x \in \mathscr{B}_G} a , \tag{3}$$

where $\mathscr{B}_G$ is a batch composed of inputs x and a is the output of the discriminator.

Mind that it is not possible to directly differentiate the output of the discriminator a with respect to the parameters of the generator, if the sequence of token probability distributions of the generator is transformed into discrete–and therefore not differentiable– tokens. Among the various options to circumvent this differentiability issue [19, 54, 52, 34, 13], we adopt the top-K softmax approach [32, 30, 31]. In short, instead of using the word vector of the most probable token at each generation step as input to the discriminator, this methodology produces approximated word vectors as a weighted average of the most likely tokens. The nearest neighbour of the approximated word vector is the most likely one as often as 98% of times (with $k = 2$), which suggests that the approximation should be good enough.

4 Batch-level GANs

Our baseline discriminator and most of the discriminators found in the literature work at the response level. While this approach has been demonstrated to be valid and useful to build more diverse open-domain dialogue models, it still presents some drawbacks. We have noticed that, sometimes, generators are able to minimise the discriminator's output by generating only a handful of relatively long and complex sentences, almost regardless of the input. One such response (more examples in Section 6.4) we have found in our experiments is: *you have no choice but to leave him, and you will never forgive him for that, and you will never forgive me.* This effect only lasts a few iterations, until the discriminator is trained to recognise those sentences as not human. However, it often happens again in other stages of the training, with different responses, which results in a more unstable and less effective learning process than desired, as we show in Section 6.

This issue cannot be easily avoided by evaluating only one response at a time, because the discriminator has no way to recognise whether the generator is generating some sentence many times (in the same training stage). It can only analyse if a response makes sense or not given the dialogue history.

We propose to extend the idea of response-level discriminators to the batch level. Our proposed batch-level discriminators combine the response-level predictions of the previously explained baseline discriminator with batch-level predictions that provide a bigger picture of the behaviour of the generator (or the nature of the real data distribution). In other words, while response-level discriminators only aim at answering the question of "*how good is this response given this previous turn?*", batch-level discriminators also tackle the question of "*how does this set of context/response pairs look like?*". This way, the generator should have more difficulties in fooling the discriminator with long and complex but similar responses. Intuitively, the batch-level discriminator could easily see that many responses in the input batch are complex but similar, and should identify that batch as generated or non-human. We show that using such discriminators improves the variability in the responses of the generator and stabilises the training.

Figure 3 shows a diagram of the proposed architecture for the batch-level discriminator. All the response-level discriminators (the Discriminative RNN Network in the figure) are the same network, i.e. they share the same parameters. The response-level contribution to the output (denoted as $a^1_{response}$ in the diagram) is computed in the same way as in the baseline discriminator (Section 3). Regarding the batch-level contribution a_{batch}, it is computed from the representations of a set of n dialogue context-response pairs. In our experiments, we found that $n = 8$ is a good enough value to produce interesting results. For simplicity and efficiency, these representations are obtained with the response-level discriminator; they are the output of the penultimate layer of the MLP in the discriminator. The sentence-level representations are processed by a standard Transformer encoder [48] without position embeddings so that its output is not affected by the order of the sequence of representations. The Transformer encoder produces n output vectors, one per input, which are averaged out. Last, a linear layer is employed to compute

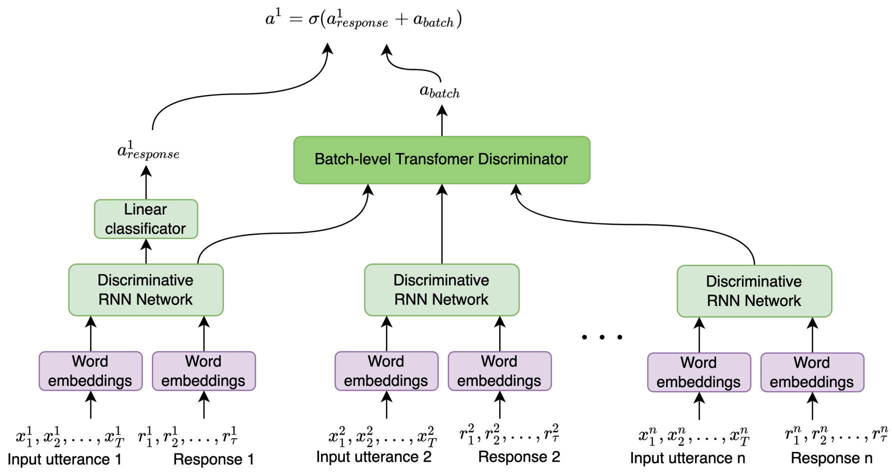


Fig. 3: Diagram of the proposed batch-level discriminator.

a_{batch}, which is added to the response-level contribution; and the sigmoid function is applied to this sum to provide the discriminator's output.

5 Experimental setup and training details

We compare the MLE baseline and the two GAN architectures (response-level and batch-level) in terms of: the variability of the generated responses, the similarity with (multiple) ground truth references (explained next in Section 5.3), and also regarding the accuracy of the discriminators. Before analysing the results, let us provide information about the training details and hyperparameters.

5.1 Corpus and preprocessing

The experiments were carried out with the English version of OpenSubtitles2018 corpus [27, 28]. As in [49], we treat each utterance as the desired output for the previous one. However, we do not consider that an utterance follows the previous one when the time difference between them is higher than three seconds. After this process, 241M input-output pairs were formed.

As for the text preprocessing, we removed uncommon and non-informative symbols and characters. We employed a BPE tokeniser [41] to tokenise the clean text. The selected size for the vocabulary was 30000. We pretrained 300-dimensional word vectors of these tokens (subwords) in the corpus, with FastText [3], and kept optimising them throughout the training.

Last, we would like to note that we did not split the corpus into any train/test partition, because the amount of training examples we process during training is significantly lower than examples in the corpus ($\sim$241M examples in the corpus vs. $\sim$77M examples sampled once during training). Thus, every example processed by any component of the GAN is *new* during training and evaluation; there are no repeated examples.

5.2 Hyper-parameter choices

The hyper-parameters of the generator and baseline discriminator are the same as in [32]. The batch-level transformer discriminator is made of two 256-dimensional layers, with four heads. Regarding the optimisation procedure, we first pretrain the generator and discriminator and then run the GAN training loop, as in [32].

We pretrained the generator during 200.000 iterations using the AdamW [33] with a learning rate of 0.001. We sampled 12.800 responses from that generator (256 every 4000 iterations) to pretrain the discriminator during 2000 iterations, with the same learning rate and optimiser. All the batches fed to the discriminator were balanced: there was a human example per each generator's example. The batches for the batch-level discriminator were split into subsets of 8 samples to compute the batch-level response evaluations. All the instances in the subsets belong to the same category (corpus or generated).

The adversarial learning loop was run 250 times. Each iteration consisted of 6 phases: 1) train the generator using adversarial learning (100 sub-iterations), 2) sample 5000 input-response pairs from the generator, 3) train the discriminator using previously sampled responses (200 sub-iterations), 4) train the generator via MLE (200 sub-iterations), 5) sample generation examples, and 6) train the discriminator (200 sub-iterations). The initial learning rate was 0.001 with a decaying factor of 0.996 when training the discriminator and the generator with the MLE criteria, and ten times smaller when training the generator in an adversarial fashion. When sampling generation examples to train the discriminator, more recently generated input-response pairs were assigned a bigger sampling probability.

5.3 Response evaluation and filtering

We measure the semantic adequacy of the generated responses via LaBSE sentence embeddings similarity [9]. We try to take into account the fact that many responses can be valid given the same dialogue history, even if they are not semantically similar. In order to find other valid responses, we first search for similar inputs in the corpus, using LaBSE embeddings too. We consider the responses to these similar inputs as valid responses to the original input of the generator, and compare its output to them. We consider the maximum value among all the comparisons as the measure of the quality of the generated response. We use a threshold of 0.8 to find similar inputs. We report the percentage of responses whose semantic similarity with the best reference is higher than the threshold.

Additionally, we take advantage of this metric to develop a heuristic to further improve the training process of GAN. In order to train the discriminator, a set of generator responses are sampled every time it is updated, and these responses are later fed into the discriminator as non-human–and therefore *bad* or non-convenient– responses. However, there are cases, especially after the generator has been trained for a while, where some of the produced responses might be completely acceptable. Using these input-output pairs as negative examples can therefore deteriorate and slow down the training. We propose to filter responses (i.e. to not use them as negative examples) with high scores in the semantic similarity metric ($>$0.8) from the corpus of the generator's responses used to train the discriminator.

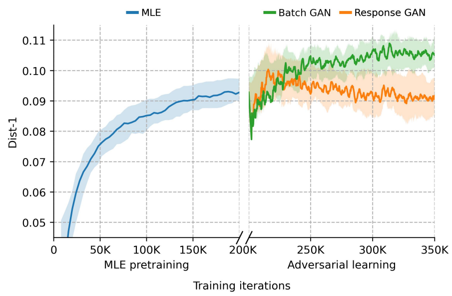
6 Results

6.1 Response variety

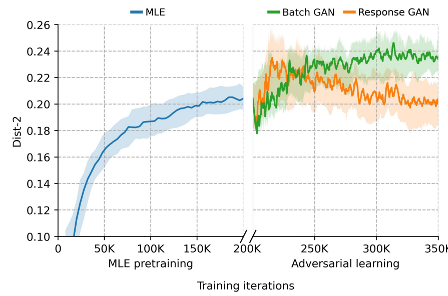
The main goal of dialogue GAN is to increase the variety in the responses of sequence-to-sequence generators. We measure the variety throughout the training process (both the MLE pretraining and the two GAN optimisation processes) with the *distinct-1* (Dist-1), *distinct-2* (Dist-2), *distinct-3* (Dist-3) and *distinct-sentences* (Dist-S) metrics [21, 36, 24]. Dist-1, Dist-2 and Dist-3 are the numbers of distinct unigrams, bigrams and three-grams (at the token level) in generated responses. The values are normalised by the total number of generated tokens to avoid favouring long sentences. On the other hand, Dist-S is the ratio of different responses. We computed the metrics with batches of 256 random inputs, and averaged them over 8 independent training runs.

The results are shown in Figure 4. The x-axis of the plots is broken in two. The first half (iterations from 0 to 200K) corresponds to the MLE pretraining, while the second half (iterations 200K to 350K) to the adversarial learning. Besides the average values after 8 training runs, we also illustrate the first and second tertiles as shaded areas, to provide information about the statistical variability of the results.

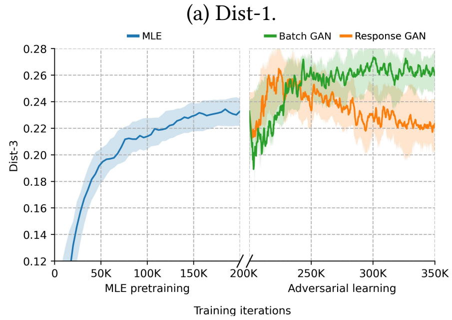
The four plots follow a similar pattern. During the first 50K iterations of the MLE pretraining stage, the variety of the responses increases highly. Then it stabilises and the improvement is less notorious until iteration 200K, when the pretraining is complete. The variety sharply drops right after the adversarial learning begins, with both the sentence and batch-level GANs. But these rapidly stabilise and start producing more and more variate responses. It is interesting that the improvement is much higher in the response-level GAN as opposed to the batch-level one. This is due to the response-level discriminator being much simpler: it can be trained



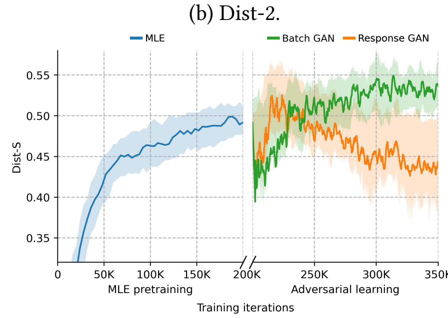
(a) Dist-1.



(b) Dist-2.



(c) Dist-3.



(d) Dist-S.

Fig. 4: Evolution of variety metrics throughout MLE pretraining and adversarial learning.

faster, but because of its aforementioned limitations its peak is lower and it even diverges (the results deteriorate with time). This is, after around 25K iterations its performance starts to decrease and it ends up with a response variety similar or lower to the MLE baseline, with greater variance. On the other hand, the batch-level discriminator processes more information and thus it takes longer to train the GAN. However, it keeps improving throughout the 150K adversarial learning iterations, and its peak is higher than the MLE baseline and the response-level GAN.

To sum up, both GAN models provide more variate responses than the MLE baseline according to the four implemented metrics. This improvement is higher and the training is more stable with batch-level GAN than with the response-level GAN. This validates our proposed batch-level discriminator.

6.2 Performance – percentage of filtered responses

As aforementioned, we also report the percentage of good responses not included as negative examples for the discriminator according to the methodology presented in Section 5.3. The evolution of this metric is shown in Figure 5.

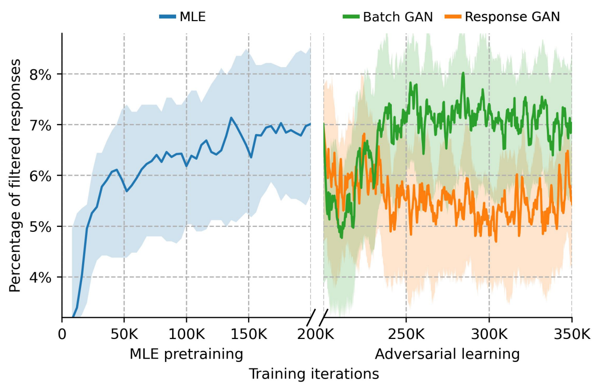


Fig. 5: Evolution of the percentage of responses not included as negative examples for the discriminator throughout the MLE pretraining and adversarial learning.

This result further validates our proposal, especially the batch-level GAN. Not only does it lead to more variate responses than the MLE baseline, but it also does so without hurting the quality of the responses. In fact, these are also more adequate semantically, even though only slightly, according to this metric. On the other hand, the response-level GAN performs slightly worse than the MLE baseline. Once again, this might be due to the intrinsic limitations of such GAN.

6.3 Discriminator accuracy

The last automatic metric we tracked was the discriminators' accuracy, shown in Figure 6. The accuracy is already fairly high from the beginning because the discriminators were pretrained with responses of the MLE baseline before computing the accuracy. The accuracy was calculated with instances unseen during training.

This plot is aligned with the learning curves shown previously. The response-level discriminator learns faster, i.e. it is able to obtain higher accuracy in the first 50K to 100K iterations (250K to 300K iterations if we start counting from the beginning of the training process, as shown in the plot). Nonetheless, the accuracy improves only slightly after the pretraining. The batch-level discriminator, as ex-

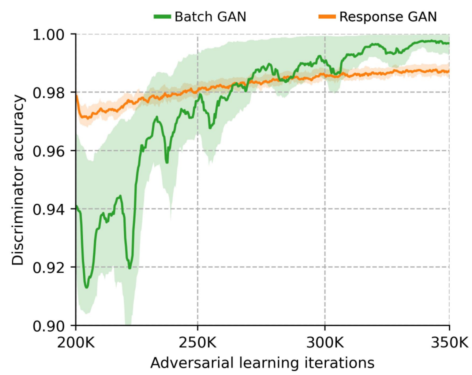


Fig. 6 Evolution of the accuracy of the discriminators throughout adversarial learning.

pected, takes longer to train, but it ends with a higher accuracy than the response-level one. This further ratifies that this kind of discriminator has a greater potential.

6.4 Generation examples

Last, we would like to show some responses that showcase the aforementioned discussed behaviour of the different models. Table 1 contains responses of four instances of the MLE baseline, response-level GAN and batch-level GAN to the same four input utterances. Each response has been generated by independent models; we use four of the eight models trained to obtain the automatic metric results to this end. We pick responses generated after the training is finished for the MLE baseline and for the batch-level GAN. As for the response-level GAN, the responses correspond to earlier stages of training, before the GAN slightly diverges, according to the automatic metrics.

The difference in the variability of the responses is quite noticeable in these examples. The MLE baseline produces generic or dull responses much more often than both GAN models, which tend to output more complex and informative sentences. Additionally, the MLE responses to the same input are more similar. This is especially visible in the case of the first input *Yesterday I saw you in the mountains*, where the four instances produced extremely similar outputs. We would also like to mention the phenomenon of the occasionally repetitive long and complex outputs of the response-level GAN. The third instance (third row) responded with *What do you want me to say, huh?* to the first and second inputs. We have noticed that this kind of behaviour occurs at many stages during the training, as previously discussed. Batch-level GAN reduce this issue significantly.

Table 1: Responses of four instances of the compared models to the same inputs.

Input utterance: *Yesterday I saw you in the mountains.*		
MLE baseline	**Response-level GAN**	**Batch-level GAN**
I was in the woods.	You were in the mountains?	You saw me, didn't you?
I was in the mountains.	You were the one who was there.	You know, you can't see me.
I was in the woods.	What do you want me to say, huh?	I was in the woods.
I saw you in the woods.	I saw you in the mountains.	You know, I've been trying to call you all day long.
Input utterance: *Could you repeat that, please?*		
MLE baseline	**Response-level GAN**	**Batch-level GAN**
I'm sorry.	I'm sorry I didn't get a chance to say goodbye.	You're a man of respect for the truth.
I'm not sure.	It's a good idea.	I'm not a doctor anymore, but I'm a writer.
I'm not going to let you do this.	What do you want me to say, huh?	And I'm sorry about your mother's death.
I'm not sure.	I'm sorry about what happened last night.	You can do better than that, sir.
Input utterance: *Tell me, what do you like?*		
MLE baseline	**Response-level GAN**	**Batch-level GAN**
I like to think you're a good man.	I like the way you look at me.	I like the way you look at them.
I don't know.	I don't know what I like about you, but I like it.	What do you like?
I like to eat.	I don't know, man, I don't think you understand, man, but…	I like to be alone.
I like the way you look at me.	I like your hair.	I like to be a little more comfortable.
Input utterance: *Could you come with us to that place?*		
MLE baseline	**Response-level GAN**	**Batch-level GAN**
I'm not sure.	I'll see if I'll have a drink with you.	What are we going to do?
I'll be right there.	Yes, of course.	I'm not going anywhere, dad.
I'm going to go.	I don't think you understand how many people you have.	I'll be right here.
I'm not sure.	I don't want to see that.	I don't want to go.

7 Conclusions

We have presented batch-level discriminators, which address some of the issues of response-level GANs. Response-level discriminators can sometimes be fooled (even if only briefly) by generators that produce only a handful of slightly long and complex sentences. Conversely, batch-level discriminators process a set of sentences, and therefore are less susceptible to misinterpreting such responses. Consequently, batch-level GANs outperform response-level GANs and the MLE baseline in terms of variability, without hurting the semantic coherence, according to our metrics. We believe that our proposal could benefit future works in GAN-related research as well as AI systems that employ discriminators in one way or another, such as GPT-2, GPT-3 or BERT-based dialogue models.

Acknowledgements This work has been partially funded by Spanish MCIU by the BEWORD project (grant number PID2021-126061OB-C42) and by the Basque Government under grant PRE 2020 1 0274.

References

1. Alex, N., Lifland, E., Tunstall, L., Thakur, A., Maham, P., Riedel, C.J., Hine, E., Ashurst, C., Sedille, P., Carlier, A., et al.: Raft: A real-world few-shot text classification benchmark. arXiv preprint arXiv:2109.14076 (2021)
2. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473 (2014)
3. Bojanowski, P., Grave, E., Joulin, A., Mikolov, T.: Enriching word vectors with subword information. arXiv preprint arXiv:1607.04606 (2016)
4. Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al.: Language models are few-shot learners. Advances in neural information processing systems **33**, 1877–1901 (2020)
5. Budzianowski, P., Vulic, I.: Hello, it's gpt-2-how can i help you? towards the use of pretrained language models for task-oriented dialogue systems. EMNLP-IJCNLP 2019 p. 15 (2019)
6. Chen, S., Wu, Y., Wang, C., Chen, Z., Chen, Z., Liu, S., Wu, J., Qian, Y., Wei, F., Li, J., et al.: Unispeech-sat: Universal speech representation learning with speaker aware pre-training. In: ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6152–6156. IEEE (2022)
7. Chowdhery, A., Narang, S., Devlin, J., Bosma, M., Mishra, G., Roberts, A., Barham, P., Chung, H.W., Sutton, C., Gehrmann, S., et al.: Palm: Scaling language modeling with pathways. arXiv preprint arXiv:2204.02311 (2022)
8. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of NAACL-HLT, pp. 4171–4186 (2019)
9. Feng, F., Yang, Y., Cer, D., Arivazhagan, N., Wang, W.: Language-agnostic bert sentence embedding. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics, pp. 878–891 (2022)
10. Ghazvininejad, M., Brockett, C., Chang, M.W., Dolan, B., Gao, J., Yih, W.t., Galley, M.: A knowledge-grounded neural conversation model. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32 (2018)
11. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Advances in neural information processing systems, pp. 2672–2680 (2014)
12. Gu, J., Tresp, V., Qin, Y.: Are vision transformers robust to patch perturbations? In: European Conference on Computer Vision, pp. 404–421. Springer (2022)
13. Gulrajani, I., Ahmed, F., Arjovsky, M., Dumoulin, V., Courville, A.C.: Improved training of wasserstein gans. Advances in neural information processing systems **30** (2017)
14. Ham, D., Lee, J.G., Jang, Y., Kim, K.E.: End-to-end neural pipeline for goal-oriented dialogue systems using GPT-2. In: Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pp. 583–592. Association for Computational Linguistics, Online (2020). DOI 10.18653/v1/2020.acl-main.54. URL https://www.aclweb.org/anthology/2020.acl-main.54

15. Hori, T., Wang, W., Koji, Y., Hori, C., Harsham, B., Hershey, J.R.: Adversarial training and decoding strategies for end-to-end neural conversation models. Computer Speech & Language **54**, 122–139 (2019)
16. Jiang, S., Ren, P., Monz, C., de Rijke, M.: Improving neural response diversity with frequency-aware cross-entropy loss. In: The World Wide Web Conference, pp. 2879–2885 (2019)
17. Karras, T., Aila, T., Laine, S., Lehtinen, J.: Progressive growing of gans for improved quality, stability, and variation. In: International Conference on Learning Representations (ICLR) (2018)
18. Komeili, M., Shuster, K., Weston, J.: Internet-augmented dialogue generation. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 8460–8478 (2022)
19. Kusner, M.J., Hernández-Lobato, J.M.: Gans for sequences of discrete elements with the gumbel-softmax distribution. arXiv preprint arXiv:1611.04051 (2016)
20. Lee, Y.J., Lim, C.G., Choi, Y., Lm, J.H., Choi, H.J.: Personachatgen: Generating personalized dialogues using gpt-3. In: Proceedings of the 1st Workshop on Customized Chat Grounding Persona and Knowledge, pp. 29–48 (2022)
21. Li, J., Galley, M., Brockett, C., Gao, J., Dolan, W.B.: A diversity-promoting objective function for neural conversation models pp. 110–119 (2016)
22. Li, J., Monroe, W., Shi, T., Jean, S., Ritter, A., Jurafsky, D.: Adversarial learning for neural dialogue generation. arXiv preprint arXiv:1701.06547 (2017)
23. Li, Y., Feng, S., Sun, B., Li, K.: Diversifying neural dialogue generation via negative distillation. In: Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 407–418 (2022). DOI 10.18653/v1/2022.naacl-main.31
24. Li, Y., Feng, S., Sun, B., Li, K.: Diversifying neural dialogue generation via negative distillation. arXiv preprint arXiv:2205.02795 (2022)
25. Li, Z., Kiseleva, J., de Rijke, M.: Improving response quality with backward reasoning in open-domain dialogue systems. In: Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 1940–1944 (2021)
26. Lin, K., Li, D., He, X., Zhang, Z., Sun, M.T.: Adversarial ranking for language generation. Advances in neural information processing systems **30** (2017)
27. Lison, P., Tiedemann, J.: Opensubtitles2016: Extracting large parallel corpora from movie and tv subtitles (2016)
28. Lison, P., Tiedemann, J., Kouylekov, M., et al.: Open subtitles 2018: Statistical rescoring of sentence alignments in large, noisy parallel corpora. In: LREC 2018, Eleventh International Conference on Language Resources and Evaluation. European Language Resources Association (ELRA) (2019)
29. Liu, P., Yuan, W., Fu, J., Jiang, Z., Hayashi, H., Neubig, G.: Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. arXiv preprint arXiv:2107.13586 (2021)
30. López Zorrilla, A., De Velasco Vázquez, M., Justo Blanco, R.: Euskaraz hitz egiten ikasten duten makina autodidaktak [basque speaking self-learning machines]. In: Proceedings of IkerGazte 2019. Udako Euskal Unibertsitatea (2019)
31. López Zorrilla, A., deVelasco Vázquez, M., Justo, R.: Euskarazko elkarrizketa sistema automatikoa sare neuronalen bidez [a neural dialogue system in basque]. Ekaia. EHUko Zientzia eta Teknologia aldizkaria (2020)
32. López Zorrilla, A., deVelasco Vázquez, M., Torres Barañano, M.I.: A differentiable generative adversarial network for open domain dialogue. In: 10th International Workshop on Spoken Dialogue Systems Technology (2019)
33. Loshchilov, I., Hutter, F.: Decoupled weight decay regularization. Computing Research Repository **arXiv:1711.05101** (2017). URL https://arxiv.org/abs/1711.05101
34. Lu, J., Kannan, A., Yang, J., Parikh, D., Batra, D.: Best of both worlds: Transferring knowledge from discriminative learning to a generative visual dialog model. In: Advances in Neural Information Processing Systems, pp. 314–324 (2017)

35. Lucas, T., Tallec, C., Ollivier, Y., Verbeek, J.: Mixed batches and symmetric discriminators for gan training. In: International Conference on Machine Learning, pp. 2844–2853. PMLR (2018)
36. Luo, T.C., Chien, J.T.: Variational dialogue generation with normalizing flows. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 7778–7782. IEEE (2021)
37. López Zorrilla, A., Torres, M.I.: A multilingual neural coaching model with enhanced long-term dialogue structure. ACM Transactions on Interactive Intelligent Systems **12**(2) (2022). DOI https://doi.org/10.1145/3487066
38. Nabeel, M., Riaz, A., Wang, Z.: Cas-gans: An approach of dialogue policy learning based on gan and rl techniques. International Journal of Advanced Computer Science and Applications **10**(7) (2019)
39. Peng, B., Zhu, C., Li, C., Li, X., Li, J., Zeng, M., Gao, J.: Few-shot natural language generation for task-oriented dialog. arXiv preprint arXiv:2002.12328 (2020)
40. Salimans, T., Goodfellow, I., Zaremba, W., Cheung, V., Radford, A., Chen, X.: Improved techniques for training gans. In: Advances in Neural Information Processing Systems, pp. 2234–2242 (2016)
41. Sennrich, R., Haddow, B., Birch, A.: Neural machine translation of rare words with subword units. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1715–1725 (2016)
42. Serban, I.V., Sordoni, A., Bengio, Y., Courville, A.C., Pineau, J.: Building end-to-end dialogue systems using generative hierarchical neural network models. In: AAAI, vol. 16, pp. 3776–3784 (2016)
43. Sordoni, A., Galley, M., Auli, M., Brockett, C., Ji, Y., Mitchell, M., Nie, J.Y., Gao, J., Dolan, B.: A neural network approach to context-sensitive generation of conversationaesponses. arXiv preprint arXiv:1506.06714 (2015)
44. Su, H., Shen, X., Hu, P., Li, W., Chen, Y.: Dialogue generation with gan. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 32 (2018)
45. Sun, Y., Hu, Y., Xing, L., Yu, J., Xie, Y.: History-adaption knowledge incorporation mechanism for multi-turn dialogue system. In: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 34, pp. 8944–8951 (2020)
46. Touvron, H., Cord, M., Douze, M., Massa, F., Sablayrolles, A., Jégou, H.: Training data-efficient image transformers & distillation through attention. In: International Conference on Machine Learning, pp. 10,347–10,357. PMLR (2021)
47. Tuan, Y.L., Lee, H.Y.: Improving conditional sequence generative adversarial networks by stepwise evaluation. IEEE/ACM Transactions on Audio, Speech, and Language Processing (2019)
48. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I.: Attention is all you need. In: Advances in neural information processing systems, pp. 5998–6008 (2017)
49. Vinyals, O., Le, Q.: A neural conversational model. arXiv preprint arXiv:1506.05869 (2015)
50. Wolf, T., Sanh, V., Chaumond, J., Delangue, C.: Transfertransfo: A transfer learning approach for neural network based conversational agents. Computing Research Repository **arXiv:1901.08149** (2019). URL https://arxiv.org/abs/1901.08149
51. Wu, S., Li, Y., Zhang, D., Zhou, Y., Wu, Z.: Diverse and informative dialogue generation with context-specific commonsense knowledge awareness. In: ACL (2020)
52. Yu, L., Zhang, W., Wang, J., Yu, Y.: Seqgan: Sequence generative adversarial nets with policy gradient. In: AAAI, pp. 2852–2858 (2017)
53. Zhang, C., Zhou, X., Wan, Y., Zheng, X., Chang, K.W., Hsieh, C.J.: Improving the adversarial robustness of nlp models by information bottleneck. In: Findings of the Association for Computational Linguistics: ACL 2022, pp. 3588–3598 (2022)
54. Zhang, Y., Gan, Z., Carin, L.: Generating text via adversarial training. In: NIPS workshop on Adversarial Training, vol. 21, pp. 21–32 (2016)
55. Zhao, T., Zhao, R., Eskenazi, M.: Learning discourse-level diversity for neural dialog models using conditional variational autoencoders. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 654–664 (2017)